Electronic Notes in Theoretical Computer Science 65 No. 6 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume65.html 16 pages

Classes of Timed Automata and the Undecidability of Universality

Arnaldo V. Moura [1,2] and Guilherme A. Pinto [3,4]

Institute of Computing, University of Campinas, C.P. 6176, 13083-970, Campinas, SP, Brazil

Abstract

Universality for deterministic Timed Automata (TA) is PSPACE-complete but becomes highly undecidable when unrestricted nondeterminism is allowed. More precisely, universality for nondeterministic TA is Π^1_1-hard and it is still open whether it is Π^1_1-complete. It is interesting to note that the entire arithmetical hierarchy is contained in this computability gap between determinism and nondeterminism. In this paper we consider three types of syntactical restrictions to nondeterministic TA, which may contribute to a better understanding of the universality problem for TA. For the first two types, which are of independent interest, the universality problem is shown to be Π^1_1-complete. For the third one, universality is Π^0_1-complete, which is the same as saying that the complementary problem is complete in the recursively enumerable class. We also show that all the restrictions define proper subclasses of the class of timed languages defined by nondeterministic TA; and establish the relationships between the classes.

1 Introduction

In [1], the universality problem for deterministic and for nondeterministic Timed Automata (TA) were shown to be, respectively, PSPACE-complete and Π^1_1-hard. The authors also reported [1, p.217] that the latter problem resisted being shown Π^1_1-complete. In this paper, we consider restrictions giving rise to classes of nondeterministic TA for which the universality problem is positioned lower in the hierarchy, while still maintaining undecidability. We are not aware of previous work on the degree of undecidability of universality for TA, besides [1]. The degree of undecidability of the reachability problem, for generalizations of TA and for hybrid systems, was investigated in [2,3,10].

[1] Email:arnaldo@ic.unicamp.br
[2] Supported by grant FAPESP, 98/05999-4
[3] Email:guialbu@ic.unicamp.br
[4] Supported by Fundação CPqD/FUNTTEL-MCT

Universality for a nondeterministic TA A has a Π^1_2 definition, with the form $\forall\rho\exists r[\varphi(A,\rho,r)]$, where the function variables ρ and r are interpreted, respectively, as timed words and runs of A over timed words, exactly as in the natural definition of the universality problem. The predicate $\varphi(A,\rho,r)$ asserts arithmetically (i.e., with quantifiers only over number variables) that "r is an accepting run of A over ρ". The question whether or not universality is Π^1_1-complete can be viewed, in this approach, as whether or not we can get rid of the quantifier $\exists r$, and assert arithmetically that "there is an accepting run r of A over ρ".

We consider general properties of infinitary automata, for which the desired arithmetical assertion follows more naturally. Three of them are listed below. Given a TA A and a timed word ρ:

(i) If a run r of A over ρ is accepting, then it makes only finitely many nondeterministic transitions;

(ii) If q is a final location appearing infinitely often in some accepting run r of A over ρ, then any finite run of A over ρ, ending in q, is a prefix of some (accepting) run of A over ρ, repeating q infinitely often;

(iii) If there is a run r of A over ρ, then r is accepting.

These properties can be enforced by syntactical restrictions and still do not lead to decidability of universality. In Sections 3 and 4, we define, respectively, *almost* deterministic TA and *pace marker* TA, which correspond to properties (i) and (ii), respectively. We show that universality for these types of TA is Π^1_1-complete. In Section 5, we consider property (iii) and define *final* TA. This property actually leads to a Π^0_1 definition and universality for final TA is shown to be Π^0_1-complete. This last result, in particular, show that the high undecidability of universality for nondeterministic TA is not exactly due to the "timed" part of the automata, or to the unrestricted nondeterminism, but to a combination of them with the traditional acceptance conditions.

In Section 6 we establish the relationships between the classes of timed languages defined by deterministic, final, almost deterministic and pace marker TA. We present a timed language for each possible intersection of these classes, and show that all three classes are proper subclasses of the class of timed languages defined by nondeterministic TA. From the perspective of expressiveness, almost deterministic TA and pace marker TA seem, intuitively, much less expressive than nondeterministic TA. Therefore, it would be somehow surprising if universality for nondeterministic TA turns out to be Π^1_1-complete also. On the other hand, it would be even more surprising if the problem were shown to be Π^1_2-complete, for this is the degree of undecidability of the universality problem for nondeterministic ω-Turing machines [4] and for recursive infinite-state ω-automata [15], which are, also intuitively, very much more expressive. The problem may also not be in Π^1_1, and not be complete for Π^1_2.

2 Preliminaries

For the sake of completeness, we recall the definitions of TA and of the arithmetical and analytical hierarchies. A *timed word* ρ, over a finite alphabet of symbols Σ, is a pair $(\overline{\sigma}, \overline{\tau})$ where $\overline{\sigma} = \sigma_1\sigma_2\cdots$ is a sequence of symbols in Σ, and $\overline{\tau} = \tau_1\tau_2\cdots$ is a strictly increasing sequence of time values $\tau_i \in \mathbb{R}_{>0}$, where for all i there is $j > i$ such that $\tau_j > i$. Let Σ^{t} denote the set of all timed words over Σ.

Given a finite set X of clock variables, a *clock constraint* δ over X is defined inductively by $\delta := x \leq c \,|\, x \geq c \,|\, \neg\delta \,|\, \delta_1 \wedge \delta_2$, where $x \in X$ and c is a positive rational constant. The set of all clock constraints over X is denoted by $\Phi(X)$. A *clock interpretation* ν for X is a function from X to $\mathbb{R}_{\geq 0}$. For $t \in \mathbb{R}$, we write $\nu + t$ for the clock interpretation which maps every clock x to $\nu(x) + t$. A clock interpretation ν for X *satisfies* a clock constraint δ over X if δ evaluates to `true` when each clock x is replaced by $\nu(x)$ in δ.

A *timed Büchi automaton* (TBA) is a tuple $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$, where: Σ is a finite alphabet of symbols; Q is a finite set of locations; $Q_0 \subseteq Q$ is a set of start locations; $F \subseteq Q$ is a set of accepting locations; X is a finite set of clocks; $T \subseteq Q \times Q \times \Sigma \times \Phi(X) \times 2^X$ is a set of transitions.

A *state sequence* r of A is a pair $(\overline{q}, \overline{\nu})$, where $\overline{q} = q_0q_1q_2\cdots$ is a sequence of locations in Q and $\overline{\nu} = \nu_0\nu_1\nu_2\cdots$ is a sequence of clock interpretations for X. A state sequence $r = (\overline{q}, \overline{\nu})$ of A is a *run* of A over a timed word $\rho = (\overline{\sigma}, \overline{\tau})$ if: $q_0 \in Q_0$, and $\nu_0(x) = 0$ for all $x \in X$; and for all $i \geq 1$, there is a transition $e = \langle q_{i-1}, q_i, \sigma_i, \delta, \lambda\rangle \in T$ such that, $(\nu_{i-1} + \tau_i - \tau_{i-1})$ satisfies δ, and $\nu_i(x) = 0$ if $x \in \lambda$, otherwise $\nu_i(x) = \nu_{i-1}(x) + \tau_i - \tau_{i-1}$ (by definition, $\tau_0 = 0$).

Given a run $r = (\overline{q}, \overline{\nu})$ over a timed word ρ, let *inf*(r) be the set of locations appearing infinitely often in r. The run r is *accepting* if *inf*$(r) \cap F \neq \emptyset$. The language $L(A)$ accepted by A is the set of all timed words ρ such that A has an accepting run over ρ. Finally, denote by $\mathcal{TBA}$ the class of all languages accepted by some TBA.

Given a TBA $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$, a location $q \in Q$ is *deterministic* if given any two distinct transitions $\langle q, q_1', a, \delta_1, \lambda_1\rangle$ and $\langle q, q_2', a, \delta_2, \lambda_2\rangle$ in T, $\delta_1 \wedge \delta_2$ is unsatisfiable. The TBA A is *deterministic* (DTBA) if $|Q_0| = 1$ and all locations in Q are deterministic. A TBA $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$ is said *complete* if given any $q \in Q$, $a \in \Sigma$ and clock interpretation ν, there is a transition $\langle q, q_1', a, \delta_1, \lambda_1\rangle \in T$ such that ν satisfies δ_1.

2.1 Segments and Concatenation

We give the natural definitions for segments and for concatenation of timed words, which will be needed in the proofs. Given a timed word ρ, its prefix of length i, $i \geq 1$, is denoted by ρ_i.

Given $\rho = (\overline{\sigma}, \overline{\tau})$, $\rho_{[i)}$ denotes the timed word which is the suffix of ρ from the i-th position on, with time values adjusted. Formally, $\rho_{[i)} = (\overline{\sigma'}, \overline{\tau'})$ where

$\sigma'_\ell = \sigma_{\ell+i-1}, \tau'_\ell = \tau_{\ell+i-1} - \tau_{i-1}$, $\ell \geq 1$. Given $\rho = (\overline{\sigma}, \overline{\tau})$ and $j, k > 0$, $\rho_{[j,k]}$ denotes the finite timed word which is the segment of ρ from position j to position k, inclusive. Formally, $\rho_{[j,k]} = \rho'_{k-j+1}$ where $\rho' = \rho_{[j)}$. If $k < j$, then $\rho_{[j,k]}$ is the *empty* timed word. Given a finite timed word ρ of length i and a timed word ρ', $\rho'' = \rho \cdot \rho'$ denotes their concatenation, that is $\rho''_i = \rho$ and $\rho''_{[i+1)} = \rho'$.

Analogous definitions hold for finite state sequences, segments and concatenation for state sequences.

2.2 The Arithmetical and Analytical Hierarchies

We recall only a few definitions. A comprehensive introduction to hierarchies of undecidability can be found in [13].

A relation $H \subseteq \mathbb{N}^2$ is *recursive* if there is a Turing machine which, for any $\langle x_1, x_2 \rangle$ given as input, stops and accepts iff $\langle x_1, x_2 \rangle \in H$. The first existential level of the arithmetical hierarchy coincides with the recursively enumerable sets, that is, $S \in \Sigma^0_1$ iff there is a recursive relation H such that $S = \{x \in \mathbb{N} \mid \exists n H(n, x)\}$, where $H(n, x)$ means $\langle n, x \rangle \in H$. The first universal level is, then, Π^0_1 where $S \in \Pi^0_1$ iff $\overline{S} \in \Sigma^0_1$. Equivalently, a set S belongs to Π^0_1 iff there is a recursive relation H such that $S = \{x \in \mathbb{N} \mid \forall n H(n, x)\}$.

Let $\mathbb{F}$ denote the set of functions from $\mathbb{N}$ to $\mathbb{N}$. A relation $H \subseteq \mathbb{F}^j \times \mathbb{N}^2$ is *recursive* if there is a Turing machine with j oracles which, for any $\langle x_1, x_2 \rangle$ given as input, stops and accepts, using oracles $f_1, f_2, \ldots, f_j$, iff $\langle f_1, f_2, \ldots, f_j, x_1, x_2 \rangle \in H$. The first levels of the analytical hierarchy are: Σ^1_1 where $S \in \Sigma^1_1$ iff there is a recursive relation H such that $S = \{x \in \mathbb{N} \mid \exists f \forall n H(f, n, x)\}$, where f is a function variable and n is a number variable; $\Pi^1_1 = \{S \subseteq \mathbb{N} \mid \overline{S} \in \Sigma^1_1\}$; and $S \in \Pi^1_2$ iff there is a recursive relation H such that $S = \{x \in \mathbb{N} \mid \forall f \exists g \forall n H(f, g, n, x)\}$, where f and g are function variables and n is a number variable.

Given two sets $A, B \subseteq \mathbb{N}$, A can be reduced to B, written $A \leq_m B$, if there is a recursive $f : \mathbb{N} \to \mathbb{N}$, such that $x \in A$ iff $f(x) \in B$. Let $\mathcal{C} \subseteq 2^{\mathbb{N}}$. A set $S \subseteq \mathbb{N}$ is $\mathcal{C}$-hard if for all $R \in \mathcal{C}$, $R \leq_m S$; and S is $\mathcal{C}$-complete if S is $\mathcal{C}$-hard and $S \in \mathcal{C}$.

2.3 Universality for $\mathcal{TBA}$

We now show that universality for TA has a Π^1_2 definition. Let $\mathcal{B}_0, \mathcal{B}_1, \ldots$ be an effective indexing of all TBAs. The universality problem is defined as the set U_{TBA} of all indices z such that for all timed words ρ, there is an accepting run r of $\mathcal{B}_z$ over ρ. First we note that it suffices to consider only rational timed words. Let $\Sigma^{\text{rt}} = \{(\overline{\sigma}, \overline{\tau}) \in \Sigma^{\text{t}} \mid \tau_i \text{ is rational}\}$. The following lemma implies that U_{TBA} is the set of all indices z such that for all rational timed words ρ, there is an accepting run r of $\mathcal{B}_z$ over ρ.

Lemma 2.1 *Let A be a TBA. Given $(\overline{\sigma},\overline{\tau}) \in \Sigma^t$, there is $(\overline{\sigma'},\overline{\tau'}) \in \Sigma^{rt}$ such that $(\overline{\sigma},\overline{\tau}) \in L(A)$ iff $(\overline{\sigma'},\overline{\tau'}) \in L(A)$.*

Proof. The intended timed word $(\overline{\sigma'},\overline{\tau'})$ can be defined inductively as in the proof of Theorem 3.17 in [1], see also [12]. □

Now we can map functions to timed words and runs. Let $\Sigma = \{a_0, a_1, \ldots, a_{k-1}\}$. We can put $\sigma_1 = a_{f(0) \pmod k}$, $\sigma_2 = a_{f(3) \pmod k}$, and so on; and $\tau_1 = (f(1)+1)/(f(2)+1)$, $\tau_2 = (f(4)+1)/(f(5)+1)$, and so on. Hence we can take two functions $d_1 : \mathbb{F} \to \Sigma^\omega$ and $d_2 : \mathbb{F} \to \mathbb{Q}^\omega$, such that $d_1(f) = \overline{\sigma}$ and $d_2(f) = \overline{\tau}$, defined as above. It is clear that for any $\rho = (\overline{\sigma},\overline{\tau})$ in Σ^{rt}, we have an $f \in \mathbb{F}$ such that $d_1(f) = \overline{\sigma}$ and $d_2(f) = \overline{\tau}$. Let d_3 and d_4 be similar functions for sequences of locations and sequences of clock interpretations, for a given TBA.

Theorem 2.2 $U_{\mathrm{TBA}} \in \Pi^1_2$.

Proof. Consider a Turing machine M_{H_1} with one oracle. M_{H_1} accepts a given pair $\langle i,j\rangle \in \mathbb{N}^2$ iff $\tau_j > i$, where τ_j is obtained by consulting the oracle according to d_2. Let $H_1 \subseteq \mathbb{F} \times \mathbb{N}^2$ be relation such that $\langle f,i,j\rangle \in H_1$ iff M_{H_1}, with oracle f, accepts the pair $\langle i,j\rangle$. Consider another Turing machine M_{H_2} with two oracles f and g which, given a tuple $\langle i,j,z\rangle \in \mathbb{N}^3$, executes the following sequence:

(i) If $j \leq i$, then rejects;

(ii) Consults the oracle f, according to d_1 and d_2, obtaining the finite timed word $\rho = (\overline{\sigma},\overline{\tau})_j$;

(iii) Consults the oracle g, according to d_3 and d_4, obtaining the finite state sequence $r = (\overline{q},\overline{\nu})_j$;

(iv) Constructs $\mathcal{B}_z = \langle \Sigma, Q, Q_0, X, T, F\rangle$;

(v) If r is a finite run of $\mathcal{B}_z$ over ρ, and $q_j \in F$, then accepts; otherwise rejects.

Take $H_2 \subseteq \mathbb{F}^2 \times \mathbb{N}^3$ as the relation such that $H_2(f,g,i,j,z)$ iff M_{H_2} accepts the tuple $\langle z,i,j\rangle$, with oracles f and g. We have that $U_{\mathrm{TBA}} = \{z \mid \forall f \exists g\, [\,(\forall i\, \exists j\, H_1(f,i,j)) \Rightarrow (\forall i\, \exists j\, H_2(f,g,i,j,z))\,]\}$. By the normal form theorem [13], $U_{\mathrm{TBA}} \in \Pi^1_2$. □

3 Almost Deterministic TBA

Almost determinism were considered before for Büchi ω-automata [6,14] as a means of achieving better complexity bounds for a probabilistic verification problem. From the perspective of expressiveness, almost deterministic and nondeterministic Büchi ω-automata define the same class of languages. The situation is more interesting for TBA. In section 6 we show that the class of timed languages defined by almost deterministic TBA lies properly between

$\mathcal{TBA}$ and $\mathcal{DTBA}$. In this section, we show that almost determinism leads to Π^1_1-completeness of universality.

Given a TBA $A = \langle \Sigma, Q, Q_0, X, T, F \rangle$ and a set $S \subseteq Q$, let $\mathrm{Reach}(S) \subseteq Q$ be the set of locations s for which there is a sequence $s_1, s_2, \ldots, s_k$, $k \geq 1$, such that $s_1 \in S$, $s_k = s$ and for every $1 \leq i < k$ there is $\langle s_i, s_{i+1}, a, \delta, \lambda \rangle$ in T. The TBA A is an *almost deterministic* TBA (ADTBA) if all locations in $\mathrm{Reach}(F)$ are deterministic.

Let A be an ADTBA. Given an oracle for a function f encoding a rational timed word ρ, if there is an accepting run $r = (\overline{q}, \overline{\nu})$ of A over ρ, then, if we are given a finite prefix of r, $(\overline{q}, \overline{\nu})_i$, such that $q_i \in F$, then, since $\mathrm{Reach}(F)$ is deterministic, the remainder of r is uniquely determined by ρ, and can be constructed by consulting the oracle. Since we can encode finite state sequences as natural numbers, we can obtain a Π^1_1 definition for the universality problem for ADTBA, U_{ADTBA}.

Theorem 3.1 $U_{\text{ADTBA}} \in \Pi^1_1$.

Proof. Let $\mathcal{A}_0, \mathcal{A}_1, \ldots$ be an effective indexing of all ADTBAs. Let d_5 be a onto function mapping numbers to finite state sequences. Consider a Turing machine M_{H_3} with one oracle which, given a tuple $\langle p, i, j, z \rangle \in \mathbb{N}^4$, behaves as follows:

(i) If $j \leq i$, then rejects;

(ii) Decodes p according to d_5 obtaining $r = (\overline{q}, \overline{\nu})_k$;

(iii) Constructs $\mathcal{A}_z = \langle \Sigma, Q, Q_0, X, T, F \rangle$;

(iv) Consults the oracle, according to d_1 and d_2, obtaining the finite timed word $\rho = (\overline{\sigma}, \overline{\tau})_k$;

(v) If r is not a run of $\mathcal{A}_z$ over ρ, or $q_k \notin F$, then rejects;

(vi) Consults the oracle again, according to d_1 and d_2, obtaining the finite timed word $\rho' = (\overline{\sigma'}, \overline{\tau'})_{k+j}$. Observe that ρ is a prefix of ρ';

(vii) Constructs the unique run $r' = (\overline{q'}, \overline{\nu'})_{k+j}$ of $\mathcal{A}_z$ over ρ', having r as a prefix (assume, w.l.o.g., that $\mathcal{A}_z$ is complete);

(viii) If $q'_{k+j} \notin F$, then rejects, otherwise accepts.

Then, $U_{\text{ADTBA}} = \{ z \mid \forall f \, [\, (\forall i \, \exists j \, H_1(f, i, j)) \Rightarrow (\exists p \, \forall i \, \exists j \, H_3(f, p, i, j, z)) \,] \}$. □

In [7], it is shown that the problem of deciding whether a nondeterministic Turing machine has an infinite computation over the empty tape that visits its start state infinitely often is Σ^1_1-complete. In [1], the complement of this problem is reduced to U_{TBA}, establishing that U_{TBA} is Π^1_1-hard. The Π^1_1-hardness of U_{ADTBA} is corollary of this result, since all the timed languages needed for the reduction can, actually, be accepted by an ADTBA. In Section 5 we will modify this reduction in order to show that universality for final TBA is Π^0_1-hard. For the sake of comparison, we recall the reduction of [1].

A nondeterministic 2-counter machine M consists of a sequence of k instructions and two counters, C and D. There are 6 types of instructions: (a) increment C and jump nondeterministically to instruction x or y; (b) decrement C and jump nondeterministically to instruction x or y; (c) if $C = 0$ jump to instruction x, otherwise jump to instruction y; (d), (e) and (f) are the same as above, exchanging D and C. A configuration of M is a tuple (i, c, d), where c and d are the current counter values, and i is the instruction to be executed. A computation of M is a sequence of related configurations beginning with $(1, 0, 0)$. A computation is recurring iff instruction 1 is executed infinitely often. Define the timed language L over the alphabet $\{b_1, b_2, \ldots, b_k, a_1, a_2\}$ such that $(\overline{\sigma}, \overline{\tau}) \in L$ iff:

(i) $\overline{\sigma} = b_{i_1} a_1^{c_1} a_2^{d_1} b_{i_2} a_1^{c_2} a_2^{d_2} \ldots$, where $(i_1, c_1, d_1)(i_2, c_2, d_2)\ldots$ is a recurring computation of M;

(ii) for all $j \geq 1$:
   (a) b_{i_j} occurs at time j;
   (b) if $c_{j+1} = c_j$, then for all a_1 at time $t \in (j, j+1)$ there is an a_1 at time $t+1$;
   (c) if $c_{j+1} = c_j + 1$, then for all a_1 at time $t \in (j+1, j+2)$ there is an a_1 at time $t-1$, except for the last one;
   (d) if $c_{j+1} = c_j - 1$, then for all a_1 at time $t \in (j, j+1)$ there is an a_1 at time $t+1$, except for the last one;
   (e) the same conditions as in (b), (c) and (d), exchanging a_2 and a_1, and exchanging d and c.

The complementary language $\overline{L}$ can be defined as a finite disjunction of several simple timed languages, which can all be accepted by ADTBAs. The disjoint union of all these ADTBAs is universal iff M *does not* have a recurring computation [12]. One example of these languages is the following. Suppose instruction 6 is of type (d). Then, $\overline{L}$ must contain the language $G_1 = \{(\overline{\sigma}, \overline{\tau}) \mid \exists i \exists j \exists k,\ i < j < k,\ \sigma_i = b_6, \sigma_j \neq a_1,\ \tau_j - \tau_i < 1,\ \sigma_k = a_1,\ \tau_k - \tau_j = 1\}$. The ADTBA C_1 in Fig. 1 accepts $L(C_1) = G_1$. In the figures, the transitions are labeled with "σ, δ, λ", meaning $\sigma = \Sigma$, $\delta =$ `true` or $\lambda = \emptyset$, when omitted.

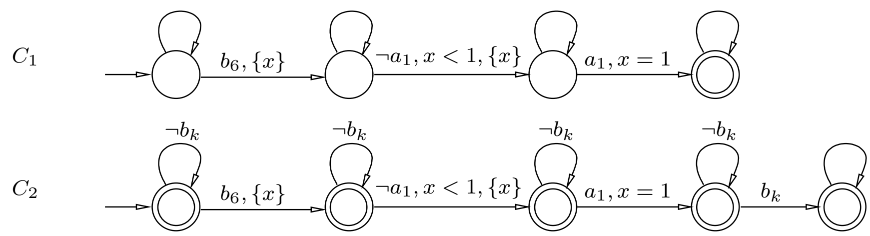

Fig. 1. Two examples of the TA needed for showing hardness of universality

4 Pace Marker TBA

Informally, we call "pace marker" an automaton in which any transition from a final location must be unconstrained and targeted to a location having an unconstrained loop transition. In this section we show that this leads to the property (ii), mentioned in the Introduction, allowing a Π^1_1 definition for universality.

A TBA $A = \langle \Sigma, Q, Q_0, X, T, F \rangle$ is a *pace marker* TBA (PMTBA) if for all $\langle q, q', a, \delta, \lambda \rangle \in T$, if $q \in F$, then $\delta = \mathtt{true}$, $\lambda = X$ and for all $b \in \Sigma$, $\langle q, q', b, \delta, \lambda \rangle \in T$ and $\langle q', q', b, \delta, \lambda \rangle \in T$.

Let A be a PMTBA, and $r = (\overline{q}, \overline{\nu})$ be a run of A over some ρ, such that there is $i < j$ and $q_i = q_j$, $q_i \in F$, $q_i \in \mathit{inf}(r)$. Note that, then, by construction, for all $x \in X$, $\nu_{j+1}(x) = 0$. It is easy to see that the following run r' is also a run of A over ρ where $q_i \in \mathit{inf}(r)$: $r' = (\overline{q'}, \overline{\nu'})$, where $q'_k = q_{j+1}$ and $\nu'_k = \nu_{j+1}$ if $i + 1 \leq k \leq j + 1$; and, otherwise, $q'_k = q_k$ and $\nu'_k = \nu_k$. That is, r' follows r until the i-th transition, goes to q_{j+1} and stays there "marking pace" until the $(j + 1)$-th symbol of ρ, when then it resumes following r.

The next theorem shows that $U_{\text{PMTBA}} \in \Pi^1_1$. Consider the following procedure for verifying whether a given rational timed word ρ is accepted by a TBA A, if we are given a final location s: consulting an oracle for a function f encoding ρ, simulate all possible finite runs of A over ρ until some run finite r_1 reaches a state $\langle s, - \rangle$; then start simulating all possible finite runs of A over the remainder of ρ, having r_1 as prefix, until some run r_2 reaches a state $\langle s, - \rangle$; and so on. If we can reach a state $\langle s, - \rangle$ infinitely often, then ρ is accepted by A. Clearly, this procedure is sound but not complete for TBAs. The theorem uses the property of the last paragraph to show that this procedure is sound and complete for PMTBA.

Theorem 4.1 $U_{\text{PMTBA}} \in \Pi^1_1$.

Proof. Let $\mathcal{A}_0, \mathcal{A}_1, \ldots$ be an effective indexing of all PMTBAs. Consider a Turing machine M_{H_4} with one oracle which, for $\langle s, i, j, z \rangle \in \mathbb{N}^4$, behaves as follows:

- (i) Constructs $\mathcal{A}_z = \langle \Sigma, Q, Q_0, X, T, F \rangle$;
- (ii) Consults the oracle, according to d_1 and d_2, obtaining the finite timed word $\rho = (\overline{\sigma}, \overline{\tau})_j$;
- (iii) Sets $S = \{\langle q, \nu \rangle \mid q \in Q_0\}$, and sets $k = 0$;
- (iv) For ℓ varying from 1 to j executes:
  - Sets $S' = \{\langle q', \nu' \rangle \mid$ there is $\langle q, \nu \rangle \in S$ and there is $\langle q, q', \sigma_\ell, \delta, \lambda \rangle \in T$ such that, $(\nu + \tau_\ell - \tau_{\ell-1})$ satisfies δ, and $\nu'(x) = 0$ if $x \in \lambda$, otherwise $\nu'(x) = \nu(x) + \tau_\ell - \tau_{\ell-1}$;
  - If there is some $\langle s, - \rangle \in S'$, then sets $k = k + 1$ and sets $S = \{\langle s, \nu^s \rangle\}$, where $\langle s, \nu^s \rangle$ if the first $\langle s, - \rangle$ in S', for some fixed order; else sets $S = S'$;

(v) If S has only one $\langle s, -\rangle$ and $k = i$, then accepts; otherwise rejects.

We have $U_{\text{PMTBA}} = \{z \mid \forall f\, [\, (\forall i\, \exists j\, H_1(f,i,j)) \Rightarrow (\exists s\, \forall i\, \exists j\, H_4(f,s,i,j,z))\,]\}$. If A_z has an accepting run $r = (\overline{q}, \overline{\nu})$ over the timed word ρ encoded by f, then s is any location in $\mathit{inf}(r) \cap F$. Assume that for some i, there is j such that $H_4(f,s,i,j,z)$. We need to show that the existence of r implies that there will, necessarily, be a j' for $i+1$ such that $H_4(f,s,i+1,j',z)$.

Let $r' = (\overline{q'}, \overline{\nu'})_j$ be the run followed by M_{H_4} for the tuple $\langle f,s,i,j,z\rangle$. Consider any m and n, $j < m < n$, such that $q_m = q_n = s$. Then A_z has the following run r'' over $\rho_{[1,n]}$: $r'' = (\overline{q''}, \overline{\nu''})_n$, where $q''_k = q_{m+1}$ and $\nu''_k = \nu_{m+1}$ if $j+1 \le k \le m+1$; $q''_k = q'_k$ and $\nu''_k = \nu'_k$ if $0 \le k \le j$; and, otherwise, $q''_k = q_k$ and $\nu''_k = \nu_k$. Therefore, $j < j' \le n$. □

The Π^1_1-hardness of U_{PMTBA} is also a corollary of the result in [1], since again all the needed timed languages can be accepted by PMTBAs. The automaton C_1 in Fig. 1, for example, is also a PMTBA. The only detail one should note is that the language L_6 defined in Section 6, which asserts that the computation of the 2-counter machine is *not* recurring, cannot be accepted by a PMTBA. Nevertheless, this language is actually not needed. The problem of asserting that a nondeterministic Turing machine has an infinite computation (not necessarily recurring) over the empty tape can also be shown Σ^1_1-complete.

We close this section noting that when a PMTBA A is marking pace it is actually insensitive to the symbols and their occurrence times. The next lemma, which will be used in Section 6, can be easily proved.

Lemma 4.2 *Given a PMTBA $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$, a timed word $\rho \in L(A)$, and a natural k, if there is a run $r = (\overline{q}, \overline{\nu})$ of A over ρ, such that $q_k \in F$ and $q_k \in \inf(r)$, then for all $\gamma \in \Sigma^t$, if $\gamma_{[1,k]} = \rho_{[1,k]}$, then for all $i > k$, there is $j > i$ such that $\gamma_{[1,j]} \cdot \rho_{[j+1)} \in L(A)$.*

5 Final TBA

Note that, given a TBA $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$, if some syntactical restriction implies property (iii) of the Introduction, that any run is accepting, then clearly the TBA $B = \langle \Sigma, Q, Q_0, X, T, Q\rangle$ has $L(B) = L(A)$. That is, the acceptance condition in A is, actually, useless for distinguishing runs. We say that a TBA $A = \langle \Sigma, Q, Q_0, X, T, F\rangle$ is a *final* TBA (FTBA) if $F = Q$.

Let us note that all three syntactical restrictions considered in this paper are not of a "timed" nature. Final TBA, in particular, are precisely the timed safety automata studied in [9,8], which are TA without acceptance conditions. While in [9,8] the interest was in the expressiveness of time safety automata, here we focus on the universality problem and, in this section, we show that universality for FTBAs is Π^0_1-complete.

On the one hand, this shows that universality for timed safety automata is still undecidable; on the other hand, that the high undecidability of universality for nondeterministic TBAs does not come from a combination of the "timed" part of the automata with unrestricted nondeterminism, but to a combination of them with acceptance conditions.

To show containment of U_{FTBA} in Π^0_1 we need only note that, given an FTBA A, if a timed word $\rho \notin L(A)$, then there is i such that there is no run of A over $\rho_{[1,i]}$. This is because the runs of A over ρ take the form of a finite number of finitely branching trees. Thus, if for all i, there is a run of A over $\rho_{[1,i]}$, then there must be an infinite run of A over ρ, which must be accepting since A is an FTBA.

Theorem 5.1 $U_{\text{FTBA}} \in \Pi^0_1$.

Proof. Let $\mathcal{A}_0, \mathcal{A}_1, \ldots$ be an effective indexing of all FTBAs. Let d_6 be a onto function mapping numbers to finite timed words. Consider a Turing machine M_{H_5} which, given a tuple $\langle i, z\rangle \in \mathbb{N}^2$, behaves as follows:

(i) Decodes i according to d_6 obtaining $\rho = (\overline{\sigma}, \overline{\tau})_k$;

(ii) Constructs $\mathcal{A}_z = \langle \Sigma, Q, Q_0, X, T, Q\rangle$;

(iii) Simulate A_z over ρ. If there is a run of A_z over ρ, accepts; otherwise, rejects.

Then $U_{\text{FTBA}} = \{z \mid \forall i\, H_5(i, z)\}$. □

Informally, we can say that the Π^1_1-hardness of U_{TBA}, U_{ADTBA} and U_{PMTBA}, is due to the fact that these automata can recognize any timed word which *does not* represent an infinite computation of a 2-counter machine. An FTBA cannot, of course, do this because $U_{\text{FTBA}} \in \Pi^0_1$. Nevertheless, an FTBA can recognize any timed word which *does not* represent a halting computation of a 2-counter machine.

Theorem 5.2 U_{FTBA} *is* Π^0_1*-hard.*

Proof. For any $C \in \Sigma^0_1$, there is a recursive H_C such that $C = \{x \in \mathbb{N} \mid \exists n\, H_C(n, x)\}$. Given $x \in \mathbb{N}$ we construct an FTBA A_x such that A_x is universal iff $x \notin C$.

Let M_x be a Turing machine (not necessarily nondeterministic) which stops, over the empty tape, iff $x \in C$. This machine needs only simulate M_{H_C} over $\langle i, x\rangle$ for increasing values of i, stopping if M_{H_C} accepts for some i.

The machine M_x can be simulated by a 2-counter Turing machine which is as described is Section 3, with the addition of one type of instruction: (g) halt. A halting computation of this machine is a sequence of related configurations ending in a configuration with the instruction halt. Assume w.l.o.g. that only the last instruction of the sequence of k instructions can be of type (g). Define the timed language L over the alphabet $\{b_1, b_2, \ldots, b_k, a_1, a_2\}$ such that $(\overline{\sigma}, \overline{\tau}) \in L$ iff:

(i) $\overline{\sigma} = b_{i_1} a_1^{c_1} a_2^{d_1} b_{i_2} a_1^{c_2} a_2^{d_2} \ldots b_{i_n} a_1^{c_n} a_2^{d_n} \ldots$, and $(i_1, c_1, d_1)(i_2, c_2, d_2) \ldots (i_n, c_n, d_n)$ is a halting computation of M_x (thus, $i_n = k$);

(ii) for all $j \leq n$, conditions (a) to (d), as in Section 3, hold.

It can be shown that the language $\overline{L}$ can be defined as a finite disjunction of timed languages which can all be accepted by FTBAs. As an example, the analog of language G_1, in Section 3, is $G_2 = \{(\overline{\sigma}, \overline{\tau}) \mid \exists i \exists j \exists k \exists \ell,\ i < j < k < \ell,\ \sigma_i = b_6, \sigma_j \neq a_1,\ \tau_j - \tau_i < 1,\ \sigma_k = a_1,\ \tau_k - \tau_j = 1,\ \sigma_\ell = b_k\}$, which is accepted by the FTBA C_2 in Fig. 1. All the other necessary languages are of a similar nature and are analogs of the languages in [12]. □

We close this section with the following lemma, whose proof is straightforward, which will be used in Section 6.

Lemma 5.3 *Given a FTBA A, if a timed word $\rho \notin L(A)$, then there is k, such that for all $\gamma \in \Sigma^t$, $\rho_{[1,k]} \cdot \gamma \notin L(A)$.*

6 Relationships between the Classes

The classes $\mathcal{ADTBA}$, $\mathcal{PMTBA}$ and $\mathcal{FTBA}$ are closed under union and intersection, but *not* under complementation [11]. Theorem 6.1 will show that there is no language in $(\mathcal{PMTBA} \cap \mathcal{FTBA}) \setminus \mathcal{ADTBA}$. The next section gives a language for any other combination. Figure 2 shows the relationships between the classes, and indicates the hardness of testing for universality.

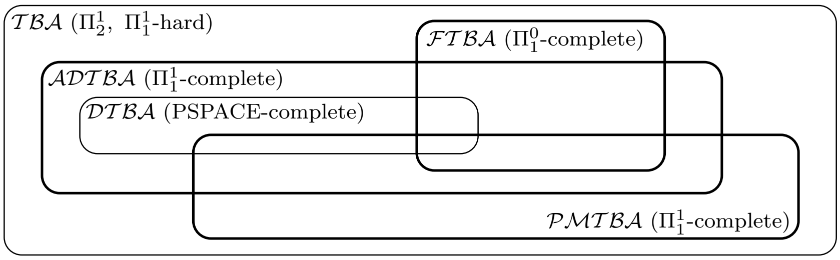

Fig. 2. Relationships between classes of timed languages

Theorem 6.1 *If $L \in (\mathcal{PMTBA} \cap \mathcal{FTBA})$, then $L \in \mathcal{ADTBA}$.*

Proof. Let $A_1 = \langle \Sigma, Q_1, Q_{01}, X_1, T_1, F_1 \rangle$ and $A_2 = \langle \Sigma, Q_2, Q_{02}, X_2, T_2, F_2 \rangle$ be, respectively a PMTBA and a FTBA, such that $L(A_1) = L(A_2) = L$. Let $G = \{q \in F_1 |$ there is $\rho \in L(A_1)$ and there is a run $r = (\overline{q}, \overline{\nu})$ of A_1 over ρ, such that $\forall i \exists j,\ j > i, q_j = q\}$.

We will show that for the following ADTBA $A = \langle \Sigma, Q, Q_{01}, X_1, T, F \rangle$, we have $L(A) = L$. Where $Q = Q_1 \cup \{q_f\}$ (disjoint union); $F = \{q_f\}$; and $T = \{\langle q, q', a, \delta, \lambda \rangle | \ [\langle q, q', a, \delta, \lambda \rangle \in T_1] \vee [q' = q_f \wedge \lambda = X_1 \wedge \delta = \texttt{true} \wedge q \in G \cup \{q_f\}]\}$. By construction of A, we can see that, if $\rho \in L(A_1)$, then $\rho \in L(A)$. We turn to the other direction.

Note that any accepting run of A must pass at least once through a location in G. Thus, if $\rho \in L(A)$, we can find a run $r = (\overline{q}, \overline{\nu})$ of A over ρ such that there is a m for which $q_m \in G$. But note that $r_{[0,m]}$ is, then, a finite run of A_1 over $\rho_{[1,m]}$.

Since $q_m \in G$, there is $\rho' \in L(A_1)$ and there is a run $r' = (\overline{q'}, \overline{\nu'})$ of A_1 over ρ', such that $\forall i \exists j,\ j > i, q'_j = q_m$. Let m' be the smallest natural such that $q'_{m'} = q_m$. Now let $\rho'' = \rho_{[1,m]} \cdot \rho'_{[m')}$. Since A_1 has the run $r'' = r_{[0,m]} \cdot r'_{[m')}$ over ρ'', we have $\rho'' \in L(A_1)$.

We now claim that $\rho \in L(A_1)$. Suppose, for the sake of contradiction, that $\rho \notin L(A_1)$. Since $L(A_1) = L(A_2)$, and A_2 is a FTBA, by Lemma 5.3, there is k, such that for all $\gamma \in \Sigma^{\mathrm{t}}$, $\rho_{[1,k]} \cdot \gamma \notin L(A_1)$.

If $k \leq m$, we have already a contradiction, since $\rho'' = \rho_{[1,k]} \cdot \rho_{[k+1,m]} \cdot \rho'_{[m')}$. If $k > m$, since $\rho_{[1,m]} = \rho''_{[1,m]}$, by Lemma 4.2, there is $\ell > k$, such that $\rho''' = \rho_{[1,\ell]} \cdot \rho''_{[\ell+1)} \in L(A_1)$. Again a contradiction, since $\rho''' = \rho_{[1,k]} \cdot \rho_{[k+1,\ell]} \cdot \rho''_{[\ell+1)}$. $\square$

6.1 A Catalog of Languages

The following list gives a timed language for every nonempty intersection of classes in Fig. 2. Table 1 below indicates with a "$\surd$" mark exactly the classes in which each language is contained. For each language, a proof that it is *not* contained in a certain class, when indicated in the table, can be given by a combination of the proof ideas of Theorems 6.2, 6.3, and 6.4 given below. For each language, one can find without much difficulty a TA of a certain class, when indicated in the table. Figure 3 gives four examples, where $L(A_i) = L_i$, L_i as in the list.

- $L_1 = \Sigma^{\mathrm{t}}$ (universal language);
- $L_2 = \{(a^\omega, \overline{\tau}) | \exists i \exists j,\ j > i,\ \tau_j = \tau_i + 1\}$;
- $L_3 = \{(a^\omega, \overline{\tau}) | \forall k \exists i \exists j,\ j > i > k,\ \tau_j = \tau_i + 1\}$;
- $L_4 = \{(\overline{\sigma} \in (b^+a)^\omega, \overline{\tau}) | \forall i\ [\ (\tau_i \in \mathbb{N} \Leftrightarrow \sigma_i = a) \wedge (\ \exists j,\ \tau_j = i) \wedge (\exists j \exists k,\ 2i < \tau_j < 2i+1,\ \tau_k = \tau_j + 1)\ ]\}$;
- $L_5 = \{(a^\omega, \overline{\tau}) | \forall i,\ \tau_i = i\}$;
- $L_6 = \{(\overline{\sigma} \in (a+b)^\omega, \overline{\tau}) | \exists i \forall j,\ j > i \Rightarrow \sigma_j = b\}$;
- $L_7 = \{(\overline{\sigma} \in (b^+a)^\omega, \overline{\tau}) | \forall i\ [\ (\tau_i \in \mathbb{N} \Leftrightarrow \sigma_i = a) \wedge (\exists j,\ \tau_j = i) \wedge (\exists j \exists k,\ k > j > i,\ \tau_k = \tau_j + 1,\ \sigma_j = \sigma_k = b)\ ]\}$;
- $L_8 = \{(\overline{\sigma}, \overline{\tau}) | \forall i \exists j,\ j > i,\ \sigma_j = a\}$;
- $L_9 = \{(a^\omega, \overline{\tau}) | \forall i\ [\ \tau_{i+1} - \tau_i \geq 1 \wedge \exists j,\ j > i,\ \tau_{j+1} - \tau_j = 1\ ]\}$;
- $L_{10} = \{(a^\omega, \overline{\tau}) | \exists i \exists j,\ \tau_j = \tau_i + 1,\ \tau_j < 2\}$;
- $L_{11} = \{(a^\omega, \overline{\tau}) | [\exists i \exists j,\ \tau_j = \tau_i + 1,\ \tau_j < 2] \wedge [\exists i ((\tau_i = 2) \wedge (\forall j,\ j \geq i \Rightarrow \tau_{j+1} = \tau_j + 2))]\}$;

Theorem 6.2 $L_3 \notin \mathcal{ADTBA}$.

Proof. We proceed by contradiction. Assume that $\mathcal{B} = \langle \Sigma, Q, Q_0, X, T, F \rangle$ is an ADTBA and that $L(\mathcal{B}) = L_3$. We first choose a special timed word $\rho^2 \in L_3$ and take any accepting run $r^2 = (\overline{q^2}, \overline{\nu^2})$ of $\mathcal{B}$ over ρ^2; then we perturb ρ^2 according to r^2, obtaining ρ^3, and show that $\mathcal{B}$ has a run $r^3 = (\overline{q^3}, \overline{\nu^3})$ over

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}
FTBA	$\surd$			$\surd$	$\surd$					$\surd$	$\surd$
PMTBA	$\surd$	$\surd$	$\surd$					$\surd$		$\surd$	
DTBA	$\surd$				$\surd$			$\surd$	$\surd$		
ADTBA	$\surd$	$\surd$			$\surd$	$\surd$		$\surd$	$\surd$	$\surd$	$\surd$

Table 1
A catalog of timed languages

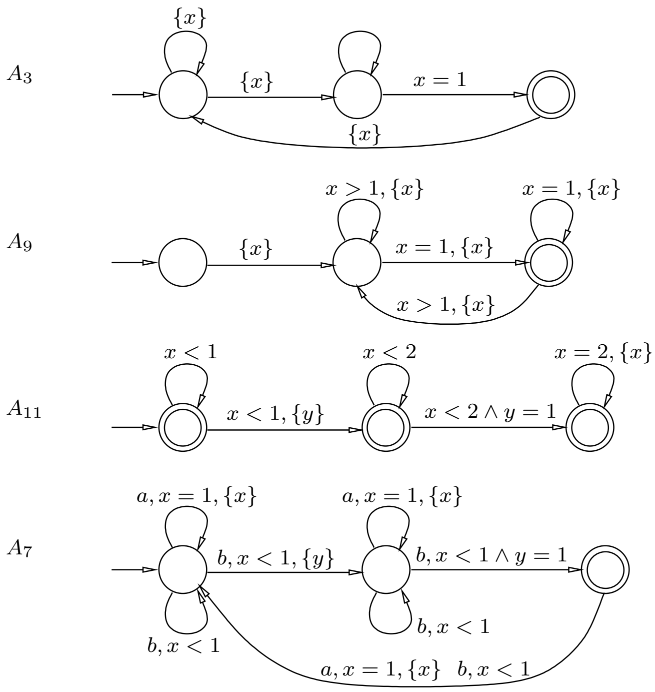

Fig. 3. Some examples of Timed Automata

ρ^3, such that $\overline{q^3} = \overline{q^2}$. The contradiction is established when we note that $\rho_3 \notin L_3$.

Let $n = |\mathrm{Reach}(F)|$ and $k = |X|$. Let C_B be a natural constant such that $C_B > 1$ and $C_B > c$ for all $c \in \mathrm{Const}(T)$, where $\mathrm{Const}(T)$ is the set of all constants that appear in some clock constraint in T. Let $\varepsilon < 1$ be a rational constant such that for all $c \in \{\mathrm{Const}(T) \cup \{1\}\}$ there is some natural m, where $c = m\varepsilon$.

In order to construct ρ^2 we define two finite timed words. Let p^0 consist of a sequence of $(nk+1)$ a's equally distributed between C_B and $C_B + \varepsilon$, that is, $p^0 = (a^{nk+1}, \overline{\tau^0})_{nk+1}$, where $\tau_i^0 = C_B + \mu i$, for $\mu = \varepsilon/(nk+2)$. The upper part of Fig. 4 illustrates p^0. Given a location $\ell \in \mathrm{Reach}(F)$ and a clock interpretation ν, since $\mathrm{Reach}(F)$ is deterministic, there is at most one finite run $(\overline{q}, \overline{\nu})_{nk+1}$ of $\mathcal{B}$ over p^0 such that $q_0 = \ell$ and $\nu_0 = \nu$; and, since there are k clocks, at least $((n-1)k+1)$ transitions in this run are such that no clock is reset *for the last time* on them. Furthermore, since the value of any clock is greater than C_B when the first a occurs, exactly the same sequence of transitions will be taken for any ν, when ℓ is fixed. Thus, there is a fixed index j, $1 \leq j \leq (nk+1)$, such that for all $\ell \in \text{Reach}(F)$ and for all ν, no clock is reset for the last time on the j-th transition of the run over p^0, starting at ℓ and ν.

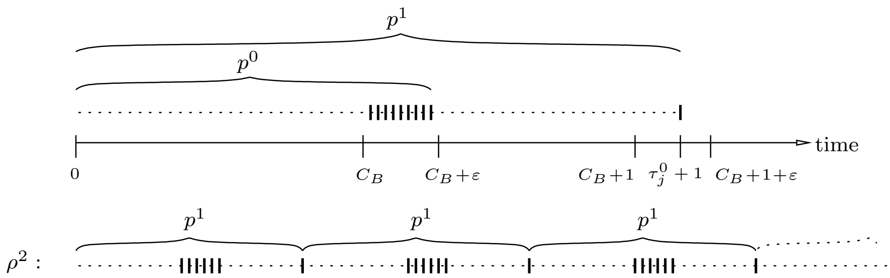

Fig. 4. Constructing the timed word ρ^2

Now, let $p^1 = (a^{nk+2}, \overline{\tau^1})_{nk+2}$ where $\tau_i^1 = \tau_i^0$ for $1 \leq i \leq (nk+1)$, and $\tau_{nk+2}^1 = \tau_j^0 + 1$. Then, $\rho^2 = (a^\omega, \overline{\tau^2})$ is the infinite concatenation of p^1, as illustrated in Fig 4. Formally, for any $i > 0$, let i^{d} and $i^{\mathrm{m}} < (nk+2)$ be naturals such that $i = i^{\mathrm{d}}(nk+2) + i^{\mathrm{m}}$. Thus, $\tau_i^2 = i^{\mathrm{d}}\tau_{nk+2}^1 + \tau_{i^{\mathrm{m}}}^1$. Define $\tau_0^1 = 0$. Clearly, $\rho^2 \in L_3$. Let $r^2 = (\overline{q^2}, \overline{\nu^2})$ be an accepting run of $\mathcal{B}$ over ρ^2. There must exist at least one such run since we assumed $L(\mathcal{B}) = L_3$. Let f be the smallest natural such that $f^{\mathrm{m}} = 0$ and $q_f^2 \in \text{Reach}(F)$. Note that r^2 is deterministic from the f-th transition on. Also, for every natural i, if $i \geq f$ and $i^{\mathrm{m}} = 0$ then for every clock $x \in X$ either x is not reset in the $(i+j)$-th transition of r^2, or x is reset in the $(i+j')$-th transition of r^2, for some j', $j < j' < nk+2$. Informally, this property makes the run r^2 insensitive to small perturbations in the occurrence times τ_i^2, for $i > f$ and $i^{\mathrm{m}} = 0$. We now obtain ρ^3 by perturbing ρ^2.

Let $\rho^3 = (a^\omega, \overline{\tau^3})$ be defined by letting $\tau_i^3 = \tau_i^2 - \mu/2$ if $i > f$ and $i^{\mathrm{m}} = 0$, otherwise $\tau_i^3 = \tau_i^2$. Thus, $\rho^3 \notin L_3$. We assume without loss of generality that $\mathcal{B}$ is complete. Let $r^3 = (\overline{q^3}, \overline{\nu^3})$ be the run of $\mathcal{B}$ over ρ^3 such that $q_i^3 = q_i^2$ and $\nu_i^3 = \nu_i^2$ for every i, $0 \leq i \leq f$. Note that there is exactly one such run, since ρ_3 equals ρ_2 up to the f-th symbol, $\mathcal{B}$ is complete and r^3 must be deterministic from the f-th transition on. We claim that r^3 and r^2 follow exactly the same sequence of transitions, which implies $\overline{q^3} = \overline{q^2}$ [12]. □

Theorem 6.3 $L_9 \notin \mathcal{PMTBA}$.

Proof. We proceed by contradiction. Assume that $\mathcal{B} = \langle \Sigma, Q, Q_0, X, T, F \rangle$ is a PMTBA and that $L(\mathcal{B}) = L_9$. Consider the timed word $\rho^2 = (a^\omega, \overline{\tau^2})$, where $\forall i, \tau_i^2 = i$. Clearly, $\rho^2 \in L_9$. Let $r^2 = (\overline{q^2}, \overline{\nu^2})$ be any accepting run of $\mathcal{B}$ over ρ^2, and let k be any natural such that $q_k^2 = q$, $q \in F$, and such that q repeats infinitely often in r^2.

Now, let $\rho^3 = (a^\omega, \overline{\tau^3})$, where $\tau^3_i = \tau^2_{i-1} + 1/2$, if $i = q+1$; and $\tau^3_i = \tau^2_i$, otherwise. Then $\rho^3 \notin L_9$, but, by Lemma 4.2, $\rho^3 \in L(\mathcal{B})$. $\Box$

Theorem 6.4 $L_9 \notin \mathcal{FTBA}$.

Proof. Again by contradiction. Assume that $\mathcal{B} = \langle \Sigma, Q, Q_0, X, \underline{T}, F\rangle$ is a FTBA and that $L(\mathcal{B}) = L_9$. Consider the timed word $\rho^2 = (a^\omega, \overline{\tau^2})$, where $\forall i, \tau^2_i = 2i$. Thus, $\rho^2 \notin L_9$.

Applying Lemma 5.3 to $\mathcal{B}$ and ρ^2, there is k, such that for all $\gamma \in \Sigma^{\mathrm{t}}$, $\rho^2_{[1,k]} \cdot \gamma \notin L(\mathcal{B})$. But take any $\rho \in L_9$. By construction of ρ^2, we have $\rho^2_{[1,k]} \cdot \rho \in L_9$. $\Box$

References

[1] Alur, R. and D. Dill, *A theory of timed automata*, Theoretical Computer Science **126** (1994), pp. 183–235.

[2] Asarin, E. and O. Maler, *Achilles and the tortoise climbing up the arithmetical hierarchy*, J. of Comp. and Sys. Sci. **57** (1998), pp. 389–398.

[3] Bournez, O., *Achilles and the tortoise climbing up the hyper-arithmetical hierarchy*, Theoretical Computer Science **210** (1999), pp. 21–71.

[4] Castro, J. and F. Cucker, *Nondeterministic ω-computations and the analytical hierarchy*, Zeitschr. f. math. Logik und Grundlagen d. Math. **35** (1989), pp. 333–342.

[5] Choueka, Y., *Theories of automata on ω-tapes: A simplified approach*, Journal of Computer and System Sciences **8** (1974), pp. 117–141.

[6] Courcoubetis, C. and M. Yannakakis, *The complexity of probabilistic verification*, Journal of the ACM **42** (1995), pp. 857–907.

[7] Harel, D., A. Pnueli and J. Stavi, *Propositional dynamic logic of nonregular programs*, J. Comput. System Sci. **26** (1983), pp. 222–243.

[8] Henzinger, T., P. Kopke and H. Wong-Toi, *The expressive power of clocks*, in: *ICALP 1995*, 1995, pp. 417–428, lNCS 944.

[9] Henzinger, T., X. Nicollin, J. Sifakis and S. Yovine, *Symbolic model checking for real-time systems*, Information and Computation **111** (1994), pp. 193–244.

[10] Miller, J., *Decidability and complexity results for timed automata and semi-linear hybrid automata*, in: *Hybrid Systems'2000* (2000), pp. 296–309, lNCS 1790.

[11] Moura, A. and G. Pinto, *Classes of timed automata and the undecidability of universality*, Technical Report IC-01-20, Institute of Computing, Univ. of Campinas (2001).

[12] Moura, A. and G. Pinto, *On almost deterministic timed automata*, Technical Report IC-01-06, Institute of Computing, Univ. of Campinas (2001).

[13] Rogers, H., "Theory of Recursive Functions and Effective Computability," The MIT Press, 1987.

[14] Safra, S., *On the complexity of ω-automata*, in: *Proc. of the 29th IEEE Foundations of Computer Science*, 1988, pp. 319–327.

[15] Sistla, A. P., *On verifying that a concurrent program satisfies a nondeterministic specification*, Information Proc. Letters **32** (1989), pp. 17–23.